



Instruction Summary for the CDP1802 COSMAC Microprocessor

RCA 1800
MICROPROCESSORS

MPM-920B

CDP1802 Register Summary

D	8 Bits	D Register (Accumulator)
DF	1 Bit	Data Flag (ALU Carry)
R	16 Bits	1 of 16 Scratchpad Registers
P	4 Bits	Designates which register is Program Counter
X	4 Bits	Designates which register is Data Pointer
N	4 Bits	Low-order Instruction Nibble
I	4 Bits	High-order Instruction Nibble
T	8 Bits	Holds old X, P after Interrupt
IE	1 Bit	Interrupt Enable
Q	1 Bit	Output Flip-flop

Interrupt Action: X and P are stored in T after executing current instruction; designator P is set to 1; designator X is set to 2; interrupt enable is reset to 0 (inhibit); and the interrupt request is serviced.

DMA Action: Finish executing current instruction. R(0) points to memory area for data transfer; data is loaded into or read out of memory; and increment R(0).

Note: In the event of concurrent DMA and INTERRUPT requests, DMA has priority.

External Flags: Four one-bit Flags set externally and tested by some branching instructions.

Information furnished by RCA is believed to be accurate and reliable. However, no responsibility is assumed by RCA for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of RCA.

CDP1802 Microprocessor Instructions

OP CODE	LEVEL1 SYNTAX	NAME	ACTION
Control Instructions			
00	IDL	IDLE	WAIT FOR DMA OR INTERRUPT; $M(R(0)) \rightarrow \text{BUS}$
C4	NOP	NO OPERATION	CONTINUE
DN	SEP reg	SET P	$N \rightarrow P$
EN	SEX reg	SET X	$N \rightarrow X$
7B	SEQ	SET Q	$1 \rightarrow Q$
7A	REQ	RESET Q	$0 \rightarrow Q$
78	SAV	SAVE	$T \rightarrow M(R(X))$
79	MARK	PUSH X, P TO STACK	$(X, P) \rightarrow T; (X, P) \rightarrow M(R(2)); \text{THEN } P \leftarrow X; R(2) - 1$
70	RET	RETURN	$M(R(X)) \rightarrow (X, P); R(X) + 1; 1 \rightarrow \text{IE}$
71	DIS	DISABLE	$M(R(X)) \rightarrow (X, P); R(X) + 1, 0 \rightarrow \text{IE}$

Memory Reference

0N	LDN reg	LOAD VIA N	$M(R(N)) \rightarrow D; \text{FOR } N \text{ NOT } 0$
4N	LDA reg	LOAD ADVANCE	$M(R(N)) \rightarrow D; R(N) + 1$
F0	LDX	LOAD VIA X	$M(R(X)) \rightarrow D$
72	LDXA	LOAD VIA X AND ADVANCE	$M(R(X)) \rightarrow D; R(X) + 1$
F8	LDI expr	LOAD IMMEDIATE	$M(R(P)) \rightarrow D; R(P) + 1$
5N	STR reg	STORE VIA N	$D \rightarrow M(R(N))$
73	STXD	STORE VIA X AND DECREMENT	$D \rightarrow M(R(X)); R(X) - 1$

Register Operations

1N	INC reg	INCREMENT REG N	$R(N) + 1$
2N	DEC reg	DECREMENT REG N	$R(N) - 1$
60	IRX	INCREMENT REG X	$R(X) + 1$
8N	GLO reg	GET LOW REG N	$R(N).0 \rightarrow D$
AN	PLO reg	PUT LOW REG N	$D \rightarrow R(N).0$
9N	GHI reg	GET HIGH REG N	$R(N).1 \rightarrow D$
BN	PHI reg	PUT HIGH REG N	$D \rightarrow R(N).1$

Logic Operations **

F1	OR		OR	$M(R(X)) \text{ OR } D \rightarrow D$
F9	ORI	expr	OR IMMEDIATE	$M(R(P)) \text{ OR } D \rightarrow D; R(P) + 1$
F3	XOR		EXCLUSIVE OR	$M(R(X)) \text{ XOR } D \rightarrow D$
FB	XRI	expr	EXCLUSIVE OR IMMEDIATE	$M(R(P)) \text{ XOR } D \rightarrow D; R(P) + 1$
F2	AND		AND	$M(R(X)) \text{ AND } D \rightarrow D$
FA	ANI	expr	AND IMMEDIATE	$M(R(P)) \text{ AND } D \rightarrow D; R(P) + 1$
F6	SHR		SHIFT RIGHT	SHIFT D RIGHT, $LSB(D) \rightarrow DF, 0 \rightarrow MSB(D)$
76	*SHRC		SHIFT RIGHT WITH CARRY	SHIFT D RIGHT, $LSB(D) \rightarrow DF, DF \rightarrow MSB(D)$
	*RSHR		RING SHIFT RIGHT	
FE	SHL		SHIFT LEFT	SHIFT D LEFT, $MSB(D) \rightarrow DF, 0 \rightarrow LSB(D)$
7E	*SHLC		SHIFT LEFT WITH CARRY	SHIFT D LEFT, $MSB(D) \rightarrow DF, DF \rightarrow LSB(D)$
	*RSHL		RING SHIFT LEFT	

Arithmetic Operations **

F4	ADD		ADD	$M(R(X)) + D \rightarrow DF, D$
FC	ADI	expr	ADD IMMEDIATE	$M(R(P)) + D \rightarrow DF, D; R(P) + 1$
74	ADC		ADD WITH CARRY	$M(R(X)) + D + DF \rightarrow DF, D;$
7C	ADCI	expr	ADD WITH CARRY, IMMEDIATE	$M(R(P)) + D + DF \rightarrow DF, D, R(P) + 1$
F5	SD		SUBTRACT D	$M(R(X)) - D \rightarrow DF, D$
FD	SDI	expr	SUBTRACT D IMMEDIATE	$M(R(P)) - D \rightarrow DF, D; R(P) + 1$
75	SDB		SUBTRACT D WITH BORROW	$M(R(X)) - D - (\text{NOT } DF) \rightarrow DF, D$
7D	SDBI	expr	SUBTRACT D WITH BORROW IMMEDIATE	$M(R(P)) - D - (\text{NOT } DF) \rightarrow DF, D; R(P) + 1$
F7	SM		SUBTRACT MEMORY	$D - M(R(X)) \rightarrow DF, D$
FF	SMI	expr	SUBTRACT MEMORY IMMEDIATE	$D - M(R(P)) \rightarrow DF, D; R(P) + 1$
77	SMB		SUBTRACT MEMORY WITH BORROW	$D - M(R(X)) - (\text{NOT } DF) \rightarrow DF, D$
7F	SMBI	expr	SUBTRACT MEMORY WITH BORROW, IMMEDIATE	$D - M(R(P)) - (\text{NOT } DF) \rightarrow DF, D; R(P) + 1$

Branch Instructions

OP CODE	LEVEL1 SYNTAX	NAME	ACTION
Short Branch			
30	BR expr	SHORT BRANCH	$M(R(P)) \rightarrow R(P) \ 0$
38	*NBR expr	NO SHORT BRANCH (SEE SKP)	$R(P) + 1$
32	BZ expr	SHORT BRANCH IF D = 0	IF D=0, $M(R(P)) \rightarrow R(P) \ 0$ ELSE $R(P) + 1$
3A	BNZ expr	SHORT BRANCH IF D NOT 0	IF D NOT 0, $M(R(P)) \rightarrow R(P) \ 0$ ELSE $R(P) + 1$
33	*BDF expr	SHORT BRANCH IF DF=1	IF DF=1, $M(R(P)) \rightarrow R(P) \ 0$
	*BPZ expr	SHORT BRANCH IF POS OR ZERO	ELSE $R(P) + 1$
	*BGE expr	SHORT BRANCH IF GREATER OR EQUAL	
3B	*BNF expr	SHORT BRANCH IF DF=0	IF DF=0, $M(R(P)) \rightarrow R(P) \ 0$
	*BM expr	SHORT BRANCH IF MINUS	ELSE $R(P) + 1$
	*BL expr	SHORT BRANCH IF LESS	
31	BQ expr	SHORT BRANCH IF Q=1	IF Q=1, $M(R(P)) \rightarrow R(P) \ 0$ ELSE $R(P) + 1$
39	BNQ expr	SHORT BRANCH IF Q=0	IF Q=0, $M(R(P)) \rightarrow R(P) \ 0$ ELSE $R(P) + 1$
34	B1 expr	SHORT BRANCH IF EF1=1	IF EF1=1, $M(R(P)) \rightarrow R(P) \ 0$ ELSE $R(P) + 1$
3C	BN1 expr	SHORT BRANCH IF EF1=0	IF EF1=0, $M(R(P)) \rightarrow R(P) \ 0$ ELSE $R(P) + 1$
35	B2 expr	SHORT BRANCH IF EF2=1	IF EF2=1, $M(R(P)) \rightarrow R(P) \ 0$, ELSE $R(P) + 1$
3D	BN2 expr	SHORT BRANCH IF EF2=0	IF EF2=0, $M(R(P)) \rightarrow R(P) \ 0$, ELSE $R(P) + 1$
36	B3 expr	SHORT BRANCH IF EF3=1	IF EF3=1, $M(R(P)) \rightarrow R(P) \ 0$, ELSE $R(P) + 1$
3E	BN3 expr	SHORT BRANCH IF EF3=0	IF EF3=0, $M(R(P)) \rightarrow R(P) \ 0$, ELSE $R(P) + 1$
37	B4 expr	SHORT BRANCH IF EF4 = 1	IF EF4=1, $M(R(P)) \rightarrow R(P) \ 0$, ELSE $R(P) + 1$
3F	BN4 expr	SHORT BRANCH IF EF4=0	IF EF4=0, $M(R(P)) \rightarrow R(P) \ 0$, ELSE $R(P) + 1$

Long Branch

C0	LBR	expr	LONG BRANCH	$M(R(P)) \rightarrow R(P).1; M(R(P) + 1) \rightarrow R(P).0$
C8	*NLBR	expr	NO LONG BRANCH (SEE LSKP)	$R(P) + 2$
C2	LBZ	expr	LONG BRANCH IF D=0	IF D=0, $M(R(P)) \rightarrow R(P).1; M(R(P) + 1) \rightarrow R(P).0$; ELSE $R(P) + 2$
CA	LBNZ	expr	LONG BRANCH IF D NOT 0	IF D NOT 0, $M(R(P)) \rightarrow R(P).1; M(R(P) + 1) \rightarrow R(P).0$; ELSE $R(P) + 2$
C3	LBDF	expr	LONG BRANCH IF DF=1	IF DF=1, $M(R(P)) \rightarrow R(P).1; M(R(P) + 1) \rightarrow R(P).0$; ELSE $R(P) + 2$
CB	LBNF	expr	LONG BRANCH IF DF=0	IF DF=0, $M(R(P)) \rightarrow R(P).1; M(R(P) + 1) \rightarrow R(P).0$; ELSE $R(P) + 2$
C1	LBO	expr	LONG BRANCH IF Q=1	IF Q=1, $M(R(P)) \rightarrow R(P).1; M(R(P) + 1) \rightarrow R(P).0$; ELSE $R(P) + 2$
C9	LBNQ	expr	LONG BRANCH IF Q=0	IF Q=0, $M(R(P)) \rightarrow R(P).1; M(R(P) + 1) \rightarrow R(P).0$; ELSE $R(P) + 2$

Skip Instructions

38	*SKP	SHORT SKIP (SEE NBR)	$R(P) + 1$
C8	*LSKP	LONG SKIP (SEE NLBR)	$R(P) + 2$
CE	LSZ	LONG SKIP IF D=0	IF D=0, $R(P) + 2$; ELSE CONTINUE
C6	LSNZ	LONG SKIP IF D NOT 0	IF D NOT 0, $R(P) + 2$; ELSE CONTINUE
CF	LSDF	LONG SKIP IF DF=1	IF DF=1, $R(P) + 2$; ELSE CONTINUE
C7	LSNF	LONG SKIP IF DF=0	IF DF=0, $R(P) + 2$; ELSE CONTINUE
CD	LSQ	LONG SKIP IF Q=1	IF Q=1, $R(P) + 2$; ELSE CONTINUE
C5	LSNQ	LONG SKIP IF Q=0	IF Q=0, $R(P) + 2$; ELSE CONTINUE
CC	LSIE	LONG SKIP IF IE=1	IF IE=1, $R(P) + 2$; ELSE CONTINUE

OP CODE	LEVEL1 SYNTAX	NAME	ACTION
------------	------------------	------	--------

Input-Output Byte Transfer

6N	OUT	dev	OUTPUT	M(R(X))→BUS; R(X) + 1; FOR N=1 TO 7
6N	INP	dev	INPUT	BUS→M(R(X)); BUS→D; FOR N=9 TO F

NOTES:

- N A HEX DIGIT
 - reg A HEX DIGIT, "R" FOLLOWED BY A HEX DIGIT, OR A SYMBOLIC NAME.
 - dev "1" THROUGH "7" OR A SYMBOLIC NAME IN THAT RANGE.
 - expr A CONSTANT, "*", OR A SYMBOLIC NAME POSSIBLY PLUS ("+") OR MINUS ("-") A CONSTANT.
 - THIS INSTRUCTION IS ASSOCIATED WITH MORE THAN ONE MNEMONIC. EACH MNEMONIC IS INDIVIDUALLY LISTED.
 - THE ARITHMETIC OPERATIONS AND THE SHIFT INSTRUCTIONS ARE THE ONLY INSTRUCTIONS THAT CAN ALTER THE DF.
- AFTER AN ADD INSTRUCTION:
- DF=1 DENOTES A CARRY HAS OCCURRED
 - DF=0 DENOTES A CARRY HAS NOT OCCURRED
- AFTER A SUBTRACT INSTRUCTION:
- DF=1 DENOTES NO BORROW; D IS A TRUE POSITIVE NUMBER
 - DF=0 DENOTES A BORROW; D IS TWO'S COMPLEMENT
- THE SYNTAX - (NOT DF) DENOTES THE SUBTRACTION OF THE BORROW

CDP1802 Instruction Summary

Op. Code	Mne-monic	Type
00	IDL	Control
0N	LDN	Mem. Ref.
1N	INC	Reg. Op.
2N	DEC	Reg. Op.
30	BR	S-Branch
31	BQ	S-Branch
32	BZ	S-Branch
33	BDF	S-Branch
34	B1	S-Branch
35	B2	S-Branch
36	B3	S-Branch
37	B4	S-Branch
38	NBR	S-Branch
	SKP	Skip
39	BNQ	S-Branch
3A	BNZ	S-Branch
3B	BNF	S-Branch
3C	BN1	S-Branch
3D	BN2	S-Branch
3E	BN3	S-Branch
3F	BN4	S-Branch
4N	LDA	Mem. Ref.
5N	STR	Mem. Ref.
60	IRX	Reg. Op.
6N*	OUT	I/O
6N*	INP	I/O
70	RET	Control

Op. Code	Mne-monic	Type
71	DIS	Control
72	LDXA	Mem. Ref.
73	STXD	Mem. Ref.
74	ADC	Arith. Op.
75	SDB	Arith. Op.
76	SHRC	Logic Op.
77	SMB	Arith. Op.
78	SAV	Control
79	MARK	Control
7A	REQ	Control
7B	SEQ	Control
7C	ADCI	Arith. Op.
7D	SDBI	Arith. Op.
7E	SHLC	Logic Op.
7F	SMBI	Arith. Op.
8N	GLO	Reg. Op.
9N	GHI	Reg. Op.
AN	PLO	Reg. Op.
BN	PHI	Reg. Op.
C0	LBR	L-Branch
C1	LBQ	L-Branch
C2	LBZ	L-Branch
C3	LBDF	L-Branch
C4	NOP	Control
C5	LSNQ	Skip
C6	LSNZ	Skip
C7	LSNF	Skip

Op. Code	Mne-monic	Type
C8	LSKP	Skip
	NLBR	L-Branch
C9	LBNQ	L-Branch
CA	LBNZ	L-Branch
CB	LBNF	L-Branch
CC	LSIE	Skip
CD	LSQ	Skip
CE	LSZ	Skip
CF	LSDF	Skip
DN	SEP	Control
EN	SEX	Control
F0	LDX	Mem. Ref.
F1	OR	Logic Op.
F2	AND	Logic Op.
F3	XOR	Logic Op.
F4	ADD	Arith. Op.
F5	SD	Arith. Op.
F6	SHR	Logic Op.
F7	SM	Arith. Op.
F8	LDI	Mem. Ref.
F9	ORI	Logic Op.
FA	ANI	Logic Op.
FB	XRI	Logic Op.
FC	ADI	Arith. Op.
FD	SDI	Arith. Op.
FE	SHL	Logic Op.
FF	SMI	Arith. Op.

*61-67 ARE OUTPUT WHILE 69-6F ARE INPUT INSTRUCTIONS

ASCII (Without Parity)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

EXAMPLES: A IS CODE 41. CARRIAGE RETURN 0D

Hexadecimal – Decimal Conversion

HEX : BINARY	HEX : DEC	HEX : DEC	HEX : DEC	HEX : DEC
0 0000	0 0	0 0	0 0	0 0
1 0001	1 4,096	1 256	1 16	1 1
2 0010	2 8,192	2 512	2 32	2 2
3 0011	3 12,288	3 768	3 48	3 3
4 0100	4 16,384	4 1,024	4 64	4 4
5 0101	5 20,480	5 1,280	5 80	5 5
6 0110	6 24,576	6 1,536	6 96	6 6
7 0111	7 28,672	7 1,792	7 112	7 7
8 1000	8 32,768	8 2,048	8 128	8 8
9 1001	9 36,864	9 2,304	9 144	9 9
A 1010	A 40,960	A 2,560	A 160	A 10
B 1011	B 45,056	B 2,816	B 176	B 11
C 1100	C 49,152	C 3,072	C 192	C 12
D 1101	D 53,248	D 3,328	D 208	D 13
E 1110	E 57,344	E 3,584	E 224	E 14
F 1111	F 61,440	F 3,840	F 240	F 15

COSMAC Development System

Utility Commands

!Maaaa xx . . .	CHANGE MEMORY AT aaaa TO xx . . .	
?Maaaa hhhh	LIST MEMORY AT aaaa FOR hhhh BYTES	NOTES:
?R	LIST CONTENTS OF REGISTERS	aaaa IS AN ADDRESS
\$Uaaaa	RUN FROM LOCATION aaaa. 2-LEVEL I/O ENABLED	xx IS A HEX DIGIT PAIR
\$Paaaa	RUN FROM LOCATION aaaa. 2-LEVEL I/O DISABLED	n IS A REGISTER NUMBER
\$Luutt	LOAD FROM DISK UNIT uu TRACK tt	h IS A HEX DIGIT

Disk Text Editor Commands

B	MOVE TO BEGINNING OF BUFFER	
Z	MOVE TO BOTTOM OF BUFFER	
nC	MOVE BY n CHARACTERS	
nL	MOVE BY n LINES	
A	APPEND TO END OF BUFFER	
nD	DELETE n LINES	
Itext\$	INSERT text	
Fsearchtext\$	FIND THE TEXT	
Ssearchtext\$Substitutetext\$	FIND AND REPLACE THE TEXT	
nX	SAVE n LINES	
G	GET THE SAVED LINES	
nT	TYPE n LINES	
nP	PARALLEL OUTPUT n LINES	
nW	DISK WRITE AND DELETE 1ST n LINES	
E	END UPDATE OF FILE	NOTE:
Y	END CREATION OF FILE	n IS A POSITIVE OR NEGATIVE NUMBER
R	DEFINE INPUT UNIT AND TRACK	RESPOND TO READ= OR WRITE= WITH uutt.
O	DEFINE OUTPUT UNIT AND TRACK	uu = 00 OR 01 tt = 00 TO 76

COSMAC Development System

Utility Subroutines

(Standard Call and Return Conventions Apply)

INIT1	83F3	A(CALL) INTO R4, A(RETURN) INTO R5 LAST RAM INTO R2, HOME DISK, SELECT GROUP 1 P=3, LBR TO 5
INIT2	83F6	SAME AS INIT1 EXCEPT LBR (R3)); R3 PREVIOUSLY INITIALIZED
READ	813E	INPUT ASCII INTO RF.1
READAH	813B	SAME AS READ. IF HEX DIGIT, THEN DIGIT SHIFTED LEFT INTO RD
TYPE6	81A2	OUTPUT ASCII CHAR AT M(R(6)). THEN INC R6
TYPE	81A4	OUTPUT ASCII CHAR IN RF.1
TYPE2	81AE	OUTPUT 2 ASCII CHARS FROM HEX DIGIT PAIR IN RF.1
LINEPR	850E	OUTPUT RF.1 TO LINE PRINTER PORT
TIMALC	80FE	READ INPUT CHAR AND SET UP TIMING CONSTANT IN RE.1. INITIALIZE RC TO POINT TO DELAY1
DELAY1	80EF	DELAY AS A FUNCTION OF M(R(3)). THEN INC R3
GOUT20	83F9	RETURN TO UT20
OSTRNG	83F0	OUTPUT ASCII STRING AT M(R(6)). DATA BYTE 00 ENDS TYPEOUT

NOTES: ALL ROUTINES EXCEPT DELAY1 USE P=3, EXIT
WITH SEP R5 AND ALTER REGISTERS X, F, DF,
RE, RF, AND M(R(2))

FOR ALL SUBROUTINES EXCEPT INIT1 AND LINEPR,
RC MUST CONTAIN AN 80EF AND RE.1 MUST CONTAIN
THE APPROPRIATE TIMING CONSTANT

Micromonitor Terminal Commands

Function	Question	Modify	Defaults	Function	Command	Defaults
Memory [▲]	?Maaaa hhhh	!Maaaa xxxx	For ?M only aaaa=0000 hhhh=1	Run, stop after hhhh breaks [▲]	\$Paaaa:hhhh	} aaaa=R(P) hhhh=1
Register [▲]	?Rn	!Rn hhhh	For ?Rn n=all For !Rn n=0 hhhh=0	Run, do hhhh instructions [▲]	\$Naaaa:hhhh	
				Run, do hhhh machine cycles [▲]	\$Saaaa:hhhh	
D [▲]	?D	!Dhh	hh=00	[For above three commands, X,P are unchanged for unspecified aaaa; X=P=0 for specified aaaa. After completion — > is displayed followed by Break Condition (M=Manual)]		
DF [▲]	?F	!Fb	b=0			
X [▲]	?X	!Xn	n=0			
P [▲]	?P	!Pn	n=0			
IE [▲]	?IE	!IEb	b=0			
T [▲]	?T	!Thh	hh=00	Return to Micromonitor Keyboard [▲] \$K		
Q [▲]	?Q	!Qb	b=0			
Wait	?W	!Wb	b=0			
Clear	?C	!Cb	b=0			
External Memory	?EM	!EMb	b=0			
				DMAIN	[Sets requests. Reset is by RR or proper S2, S3 response]	!DI
				DMAOUT	[half- duplex mode]	!DO/ !CH
					[full- duplex mode]	!DO/ !CF
				Interrupt		!I
				Reset Requests		!RR

MOPS Command Summary List

STI	Set the terminal as the input device
STO	Set the terminal as the output device
\$DKI uutt	Set the disk as the input device
\$DKO uutt	Set the disk as the output device
\$DKC	Close a disk file
\$O uutt	Set both terminal and disk as output devices
?CPU	Dump the CPU state to the output device(s)
?MRn hhhh	Dump memory pointed to by register n for hhhh bytes
\$Haaa ₁ aaa ₂ uutt	Hold the state of the machine on disk
\$L uutt	Load a disk file
\$MSG	Type a message to the terminal
\$WB	Wait for a break condition
\$WT dddd	Wait for dddd (decimal) seconds
◆ \$MB	Force a manual break
!J hhhh	Set the index J to hhhh
!J+	Increment the index J
!J-	Decrement the index J
?J	Question the current value of the index J
◆ !BP aaa . . .	Declare software breakpoints on executable addresses
◆ !BR aaa . . .	Remove breakpoint addresses from current list
◆ ?BP	Question the current software breakpoint addresses
\$GO TO %label	Search forward for %label and if found continue at that point, else end
\$IF v ₁ op v ₂ THEN GO TO %label	If the relation is true, execute "go to", else proceed
\$DKW uutt	Write a command sequence to a disk file
\$DKL uutt	List a disk file to the terminal
\$DKP uutt	Print a disk file to the line printer
\$U	Return to the utility program (UT20)

NOTES:

aaaa is an address

aaa₁ is starting address

aaa₂ is ending address

uutt denotes unit uu, track tt

d is a decimal digit

label is an alphanumeric label

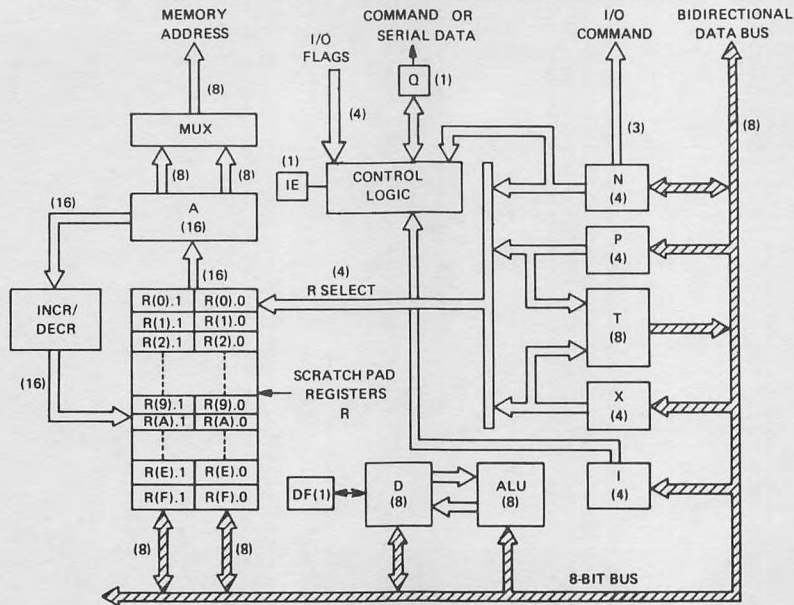
◆ In MOPS version 2.0 requiring an 8K system.

v is one of [Rn, D, F, X, P, IE, T, Q, W, C,

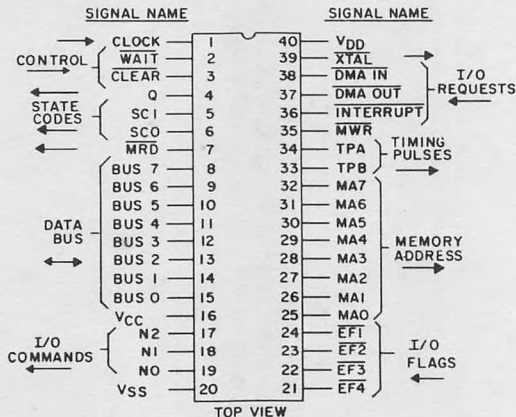
EM, IR, E_{ff}, I_p, J, #hhhh, PROMPT, . . , %]

op is one of [=, >, <, EQ, LT, GT, LE, GE, NE]

Data Flow in RCA-CDP1802 COSMAC Microprocessor



Terminal Assignments for the RCA CDP1802 COSMAC Microprocessor



92CS-27467R1

RCA Solid State

Brussels · Buenos Aires · Hamburg · Madrid · Mexico City · Milan
Montreal · Paris · Sao Paulo · Somerville NJ · Stockholm
Sunbury-on-Thames · Taipei · Tokyo